

## CPSC 1155 – Lab 6

### Math, Characters, Strings, Files

#### Lab Introduction

This lab helps you practice with Math functions, characters, strings, and files in C++.

#### Learning Objectives

At the end of this lab, you should be able to:

- Write mathematical equations in C++
- Use characters and strings in C++ programs
- Read / write from / to files

#### Lab Readings

Chapter 4 – Math, Characters, Strings

#### Lab Instructions

For each **Problem Statement**, follow the steps below:

1. Read the problem statement and clarify the problem.
  - a. Break the problem into smaller problems if needed.
2. Determine the IPO.
  - a. Determine input, output, intermediate variables, constants, and conditions.
  - b. Declare the variables and constants (data type + meaningful names).
  - c. Work out the problem by hand using typical input values. Determine the range of valid input values.
  - d. Determine the process.
3. Write a pseudocode as required.
4. Write a C++ program (use the given filename) that implements the pseudocode.
  - a. Add comments where needed. Make sure to use a comments header to reflect the intention of your program and name of the author (you) and the date the program was written.
  - b. Test, debug, and execute the program using typical values.

Submit according to the instruction in the "Lab Submission" section.

#### Problem Statements

1. [5] (mathOutput.cpp) Write a **C++ program** that inputs a number as x, calculates the following expression k using C++ mathematical functions, and displays the results with 6 digits after the decimal point. You need to include `<cmath>` library.

$$k = e^{-x} + \sqrt{e^{-x}} + e^{-\sqrt{x}} + \frac{e^x}{x}$$

Here is a sample run:

```
Enter x: 2.0
k = 4.440860
```

Determine the limitations on input value to ensure valid output values. The program tests for the validity of the input.

To display 6 digits after the decimal point, you may use the following statement (you need to include `<iomanip>` library).

```
cout << fixed << setprecision(6) << k;
```

2. [6] (charAscii.cpp) Write a **C++ program** that:

- a. Receives an 8-bit ASCII code (an integer between 0 and 255) and displays the equivalent character. Here is a sample run:

```
Enter an ASCII code (between 0 and 255): 69
The character is E

Enter an ASCII code: 300
Invalid entry
```

- b. Receives a character, checks if it is a lowercase or uppercase letter, and displays its equivalent ASCII code in decimal. Here is a sample run:

```
Enter a letter (A to Z or a to z): E
E is uppercase
The ASCII code for E is 69

Enter a letter (A to Z or a to z): %
Invalid entry
```

- c. Generates and displays two random letters; one lowercase and one uppercase.

3. [5] (strTable.cpp) Write a **C++ program** that reads the first name, last name, and grade for a student and displays the results in a tabular form as shown below (the column width is 12 and the text is left-aligned). The first row is the heading. The program asks the user to enter values for the second row. Use `getline` to read the space in a string correctly. The program tests the validity of the grade (between 0 and 100). You need to include `<string>` library.

Here is a sample run:

| First Name | Last Name | Grade |
|------------|-----------|-------|
| Mary Ann   | Lu        | 85.5  |

To manipulate text, you may use the following statement (you need to include `<iomanip>` library):

```
cout << left << setw(12) << "First Name";
```

4. [6] (fileIO.cpp) We want to write a **C++ program** to write / read data to / from a file. You need to include `<fstream>` library.

- a. The first part of the program writes the grades for 3 students to a local file.

Declare a variable of type `ofstream` which is used to output a stream into a file:

```
ofstream output; // output is the name of the variable
```

Use the following function (`open`) to create a text file on your local drive:

```
output.open("grades.txt"); // grades.txt is the name of the file
```

Prompt the user to input values for the grades of 3 students.

Use the output operator (<<) to write the grades into grades.txt:

```
output << grade1 << " " << grade2 << " " << grade3 << endl;
```

You may use IO manipulation commands (e.g. setw) in the above statement.

Close your file:

```
output.close();
```

After running the code, locate the grades.txt file on your computer. It must show the data you entered.

- b. The second part of the program reads the grades from the local file.

Declare a variable of type ifstream which is used to input a stream from a file:

```
ifstream input;          // input is the name of the variable
```

Use the following function (open) to open a text file on your local drive:

```
input.open("grades.txt");      // grades.txt is the name of the file
```

Use the input operator (>>) to read data from the above file:

```
input >> score1 >> score2 << score3;
// the values on the file will be stored in the above variables
```

Use cout to display the above values.

Close your file:

```
input.close();
```

### Lab Submission

Submit a zip folder named as yourName\_Lab6.zip to Brightspace. This folder should consist of **C++ codes** in individual .cpp files.

Please make sure that all your .cpp files compile and run properly before submission. Your file must run properly in order to receive full marks.

### Marking Scheme

The marks are given in square brackets [ ] for each question.